Introdução à Robótica com Arduino
Desvio de Obstáculos

Michelle Mendes Santos

MANUAL DO ALUNO

INSTITUTO FEDERAL
Minas Gerais

LABORATÓRIO DE INOVAÇÃO COLABORATIVO

1.1. O que é a Robótica?

Robótica é um ramo da ciência que engloba mecânica, eletroeletrônica e computação, responsável pelo estudo e pela construção de sistemas capazes de realizar tarefas de maneira autônoma (seguindo uma programação) ou o controle humano. A Robótica refere-se ao estudo dos robôs.

A Robótica se divide em duas grandes áreas: Robótica de Manipuladores e Robótica Móvel. Os manipuladores robóticos são robôs que, normalmente, são fixos em um ponto e, por meio de seus elos e juntas, podem se movimentar para alcançar objetos em torno de si e realizar tarefas específicas em um ambiente estruturado. A Figura 1 apresenta exemplos de robôs manipuladores.

Figura 1: Exemplos de Manipuladores Robóticos
Fonte: http://www.autoentusiastasclassic.com.br/2013/02/as-leis-da-robotica-e-o-carro-autonomo.html

A Robótica Móvel se destina ao estudo dos robôs que podem se locomover no ambiente, seja ele externo ou interno. Os robôs móveis podem sem terrestres, aéreos, ou aquáticos e, para cada ambiente, ele deve contar com um sistema de locomoção adequado a esse ambiente. A figura 2 apresenta alguns exemplos de robôs móveis.

Figura 2: Exemplos de Robôs Móveis
Fonte: http://graduacao.ufabc.edu.br/eiar/conteudo/ensino/disciplinas/Robotica/FundamentosRobotica.html

O robô que será desenvolvido nesta atividade é um robô móvel terrestre e deverá realizar sua tarefa sem a intervenção humana, seguindo uma programação prévia. Esse tipo de robô é conhecido como robô autônomo.

Você já parou para pensar por que os robôs funcionam? Para que um veículo seja considerado um robô, ele deve possuir alguns sistemas que irão conferir a ele diversas habilidades. Um robô móvel autônomo deve possuir, pelo menos um sistema de locomoção, sensores para conhecer o ambiente e seu próprio estado e um sistema de processamento das informações, que irá interagir com seus componentes e tomar a decisão mais acertada de acordo com sua programação. Para funcionar de maneira autônoma, o robô precisa ser equipado com Sensores, que enviarão informações a um Controlador; este processará a lógica de programação e tomará decisões, enviando comandos aos Atuadores, que farão com que as ações sejam executadas, como seguir em frente, ou virar, por exemplo. Esse esquema pode ser observado na figura 3.

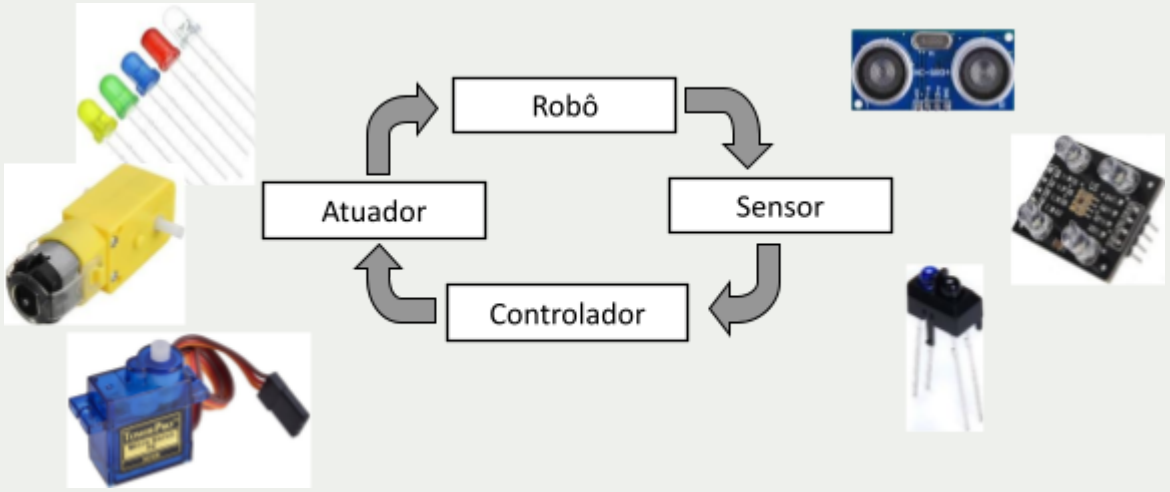


Figura 3: Esquema básico de funcionamento de um robô autônomo
Fonte: Autoria própria

É importante lembrar que, para que todos esses componentes possam funcionar corretamente, é necessário que exista uma fonte de energia. No caso de robôs móveis, a energia deve ser fornecida por meio de baterias, uma vez que eles não podem estar presos a fios. A figura 4 apresenta um esquema dos principais componentes dos robôs.

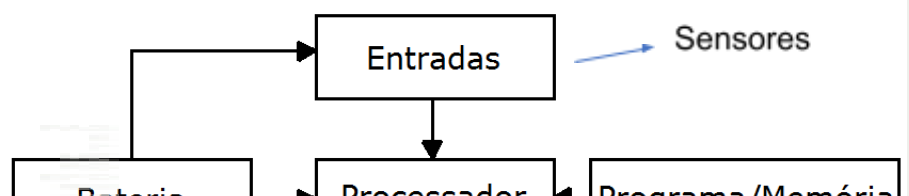


Figura 4: Principais componentes de um robô autônomo
Fonte: Autoria própria

1.2. Situação problema

Você já deve ter visto ou se deparado com um robô aspirador de pó, que faz uma trajetória dentro de um ambiente fechado e vai sugando a sujeira para o seu interior. Você também já deve ter assistido a algum vídeo que apresenta carros que “dirigem sozinhos” não é mesmo? São os “carros autônomos”. Já parou para pensar em como esse tipo de robô (autônomo) se locomove dentro dos espaços e, ao encontrar um obstáculo, como um móvel da casa, uma parede, ou até um pedestre, ele muda de direção e continua realizando sua tarefa? Para isso, esse tipo de robô precisa de sensores, que detectam a presença de um obstáculo à sua frente, e de uma lógica de programação, que faz com que ele desvie sua trajetória sempre que encontrar esse obstáculo.

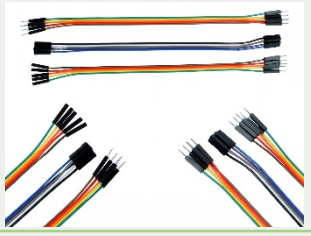
Na atividade prática desta oficina, você irá entender o funcionamento de um sensor ultrassônico e como ele detecta a presença de obstáculos à sua frente. Você irá realizar sua conexão com o controlador Arduino e elaborar uma lógica de controle que comanda as rodas do robô a fim de andar para frente até que algum obstáculo esteja a uma distância pré-determinada e, então, girar no sentido horário ou anti-horário, desviando do mesmo.

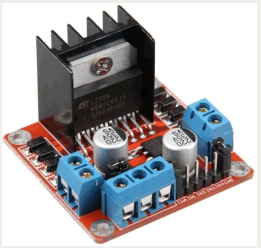
1.3. Questão motriz

Ao final desta atividade, você será capaz de responder à pergunta: “Como um robô autônomo pode identificar e desviar de um obstáculo em seu caminho?” e ainda vivenciar na prática os desafios da robótica móvel e dos robôs autônomos.

Materiais necessários

Para a realização desta atividade, são necessários os materiais listados na tabela a seguir:

Item	Nome do item	Descrição	Quant.	Foto	Substituição
1	Kit Chassi 2WD ou 4WD	Kit de Chassi em acrílico, MDF ou similar, rodas e motores, parafusos e suportes para montagem do robô	1		
2	Sensor Ultrassônico	Sensor Ultrassônico para Arduino HC-SR04	1		Não há substituto
3	Arduino UNO	Placa Controladora Arduino UNO ou compatível com cabo USB	1		Arduino Mega
4	Bateria 9V	Bateria 9V, preferencialmente recarregável e previamente recarregada	1		Bateria 9V não-recarregável
5	Jumpers	Conjunto de jumpers Macho-fêmea e Macho-macho	1		Pequenos fios feitos com cabos de rede, com as pontas decapadas
6	Clipe Bateria 9V	Clipe para bateria de 9V com pino P4	1		Não há substituto
7	Suporte Sensor Ultrassônico	Suporte em acrílico, MDF, impressão 3D ou similar para encaixar o sensor ultrassônico	1		Não há substituto

8	Módulo Ponte H	Driver L298n de Ponte H dupla	1		Módulo Ponte H Dupla HG7881

Atenção: Você receberá os materiais já montados, com a estrutura mecânica pronta para uso, restando apenas realizar as conexões com o uso dos jumpers.

2.1 O que é o Arduino

Segundo o site oficial do projeto, “Arduino é uma plataforma open-source de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobistas e qualquer pessoa interessada em criar objetos ou ambientes interativos” (Arduino, 2022) ou seja, o Arduino é uma plataforma formada por dois componentes: A placa, que é o Hardware usado para construir os projetos e a IDE Arduino, que é o Software utilizado para programar o que se deseja que a placa faça.

A placa Arduino a ser utilizada nesta oficina é o UNO, que é o modelo mais utilizado na robótica educacional (figura 5). Ela possui um Microcontrolador, que é o cérebro da placa – é onde a lógica de programação será processada para que as decisões sejam tomadas automaticamente. Possui também pinos digitais e analógicos, que possibilitam a conexão dos sensores e atuadores à placa. É possível utilizar seus pinos de alimentação para fornecer energia aos sensores e atuadores que serão ligados a ela. Para realizar a programação, pode-se utilizar seu conector USB que, por meio de um cabo, será conectado a um computador e a programação será enviada para ser armazenada em sua memória. Esse conector também é capaz de alimentar os componentes da placa, quando conectado a um computador. Porém, quando se necessita de mobilidade, é preciso desconectar a placa do computador e utilizar uma alimentação independente. Para alimentar o Arduino externamente, existe um conector no modelo P4 para fornecimento de energia.

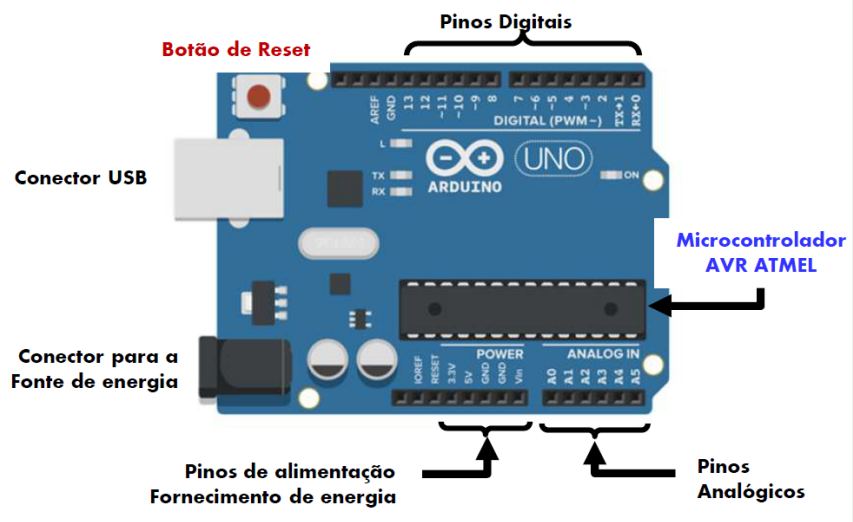


Figura 5: Principais componentes de uma placa Arduino UNO
Fonte: Autoria própria

Para realizar a programação da placa, é necessário utilizar o software Arduino IDE, disponível para download gratuito no site oficial do projeto (https://www.arduino.cc/). A interface do Arduino IDE pode ser vista na Figura 6.

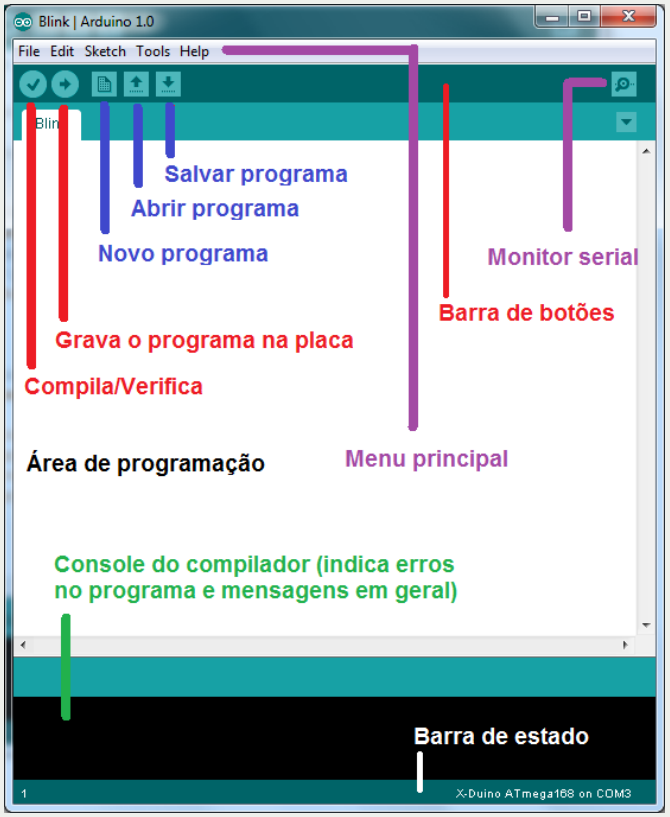


Figura 6: Interface do Software Arduino IDE.
Fonte: http://www.professorakeila.com.br/2018/04/entendendo-o-arduino-ide.html

Nesse software, deve-se escrever a programação, basicamente, em três blocos. O primeiro bloco é aquele em que se escreve aqueles comandos que não ficam dentro de nenhuma função – são definições, configurações, inclusão de bibliotecas, etc. O segundo

bloco é a função *void setup*. Nela são definidos os pinos do Arduino que serão utilizados e como serão utilizados (se como entradas ou saídas). Também são escritos nesse bloco aqueles comandos que devem ser executados apenas uma vez ao início da execução do programa. O terceiro bloco é a função *void loop*. Nela são inseridos os comandos que serão executados como um loop, ou seja, continuamente a partir do momento em que a lógica for carregada para a placa. A figura 7 apresenta um exemplo de programação contendo esses três blocos.

Figura 7: Exemplo de programa Arduino IDE
Fonte: Autoria própria

Após realizar a programação na interface, deve-se selecionar o modelo de placa (Fig. 8) e a Porta Serial (Fig. 9), no menu Ferramentas (Tools). Deve-se, então, compilar o código para verificar se há erros de digitação e, caso esteja tudo correto, deve-se carregar o programa na placa.

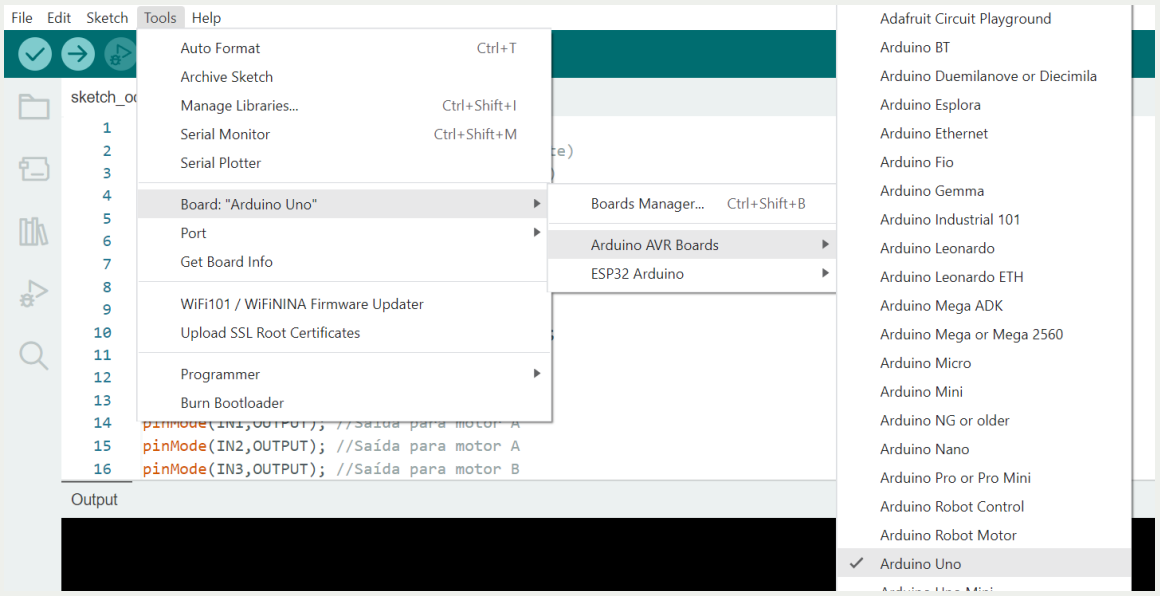


Figura 8: Seleção do Modelo de Placa Arduino UNO
Fonte: Autoria própria

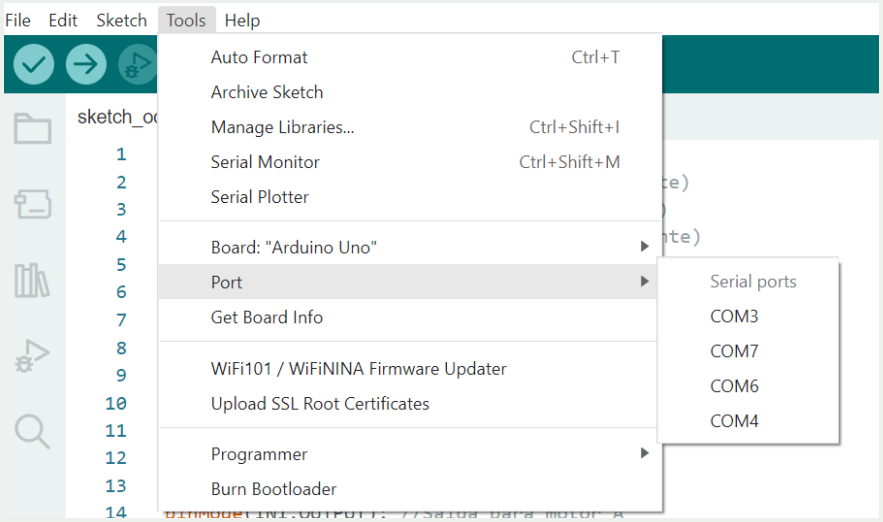


Figura 9: Seleção da Porta Serial
Fonte: Autoria própria

Caso esteja interessado em se aprofundar um pouco mais no estudo do Arduino, existem diversos sites que trazem tutoriais e apostilas de como começar a programar nessa plataforma. Algumas referências podem ser encontradas em (Mota, 2021).

Mídia digital: Links de vídeos disponíveis na internet com uma introdução ao Arduino.
- Curso “Introdução ao Arduino, o básico para começar”, da UFRGS (https://lumina.ufrgs.br/course/view.php?id=30)
- Canal “Brincando com Ideias”, no YouTube (https://www.youtube.com/c/BrincandocomIdeias)

2.2 Módulo Ponte H

O acionamento de um motor de corrente contínua, ou motor cc, é feito de maneira muito simples: quando energizado com uma polaridade, o motor gira em um sentido e ao inverter a polaridade da alimentação, ele gira no sentido contrário. Dessa maneira, para fazer as rodas do robô girarem para frente e para trás, bastaria inverter a polaridade do motor ao longo do movimento, correto? Sim! Mas como trocar os fios do motor de lugar enquanto ele se movimenta? E como fazer isso automaticamente, sem realizar mudanças físicas no robô? É nessa tarefa que entra o módulo Ponte H.

Uma Ponte H funciona como um conjunto de chaves eletrônicas, que irão abrir ou fechar de acordo com um comando que o controlador irá enviar. Dessa maneira, o sentido do motor pode ser invertido pelo acionamento dessas chaves. A figura 10 apresenta o circuito que esquematiza o funcionamento de uma Ponte H. Nessa figura, pode-se

observar que quando as chaves 1 e 4 estão conectadas, a alimentação positiva está do lado esquerdo do motor e a negativa está do lado direito. Quando as chaves 2 e 3 estão fechadas, a alimentação vista pelo motor está invertida, com o pólo positivo do lado direito do motor e o pólo negativo do lado esquerdo. Dessa forma, pode-se inverter o sentido de giro do motor apenas abrindo e fechando as chaves corretas.

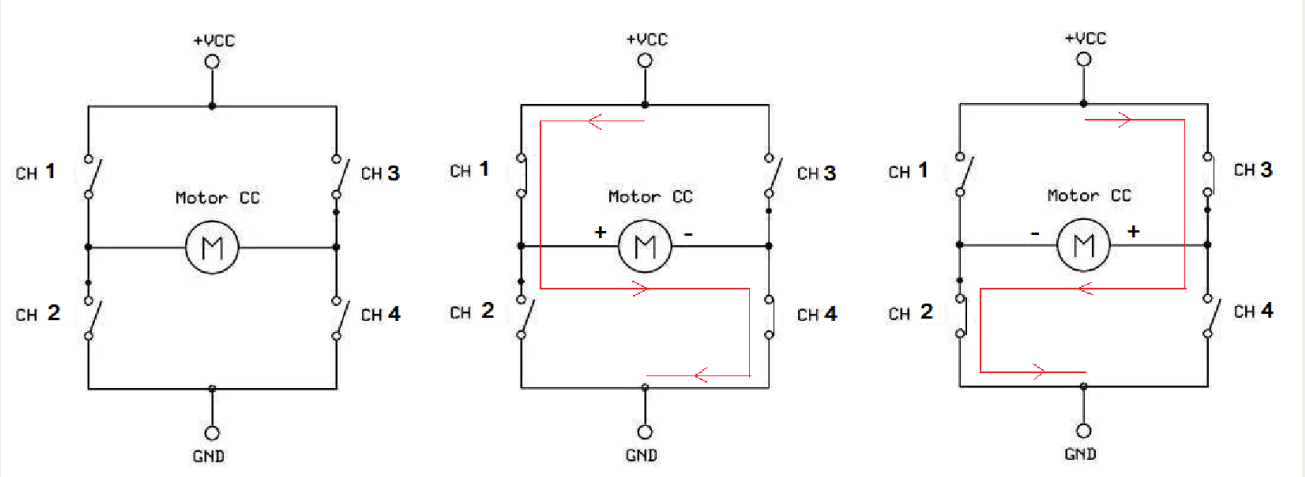


Figura 10: Circuito esquemático do funcionamento da ponte H.
Fonte: Autoria própria

O módulo Ponte H dupla utilizado nesta aplicação possui dois circuitos de chaves eletrônicas conectadas em Ponte H, o que significa que se pode acionar dois motores e inverter suas polaridades com apenas um módulo. A figura 11 apresenta uma foto do Módulo Ponte H utilizado nesta atividade.

Nesse módulo, é necessário conectar os motores do robô aos bornes destacados como “MOTOR A” e “MOTOR B”, ligar a alimentação nos bornes correspondentes, dependendo de seu valor, e conectar as entradas do módulo aos pinos digitais do Arduino.

O detalhamento das conexões necessárias nesta atividade são encontrados nas seções 3.1 e 3.2 deste caderno.

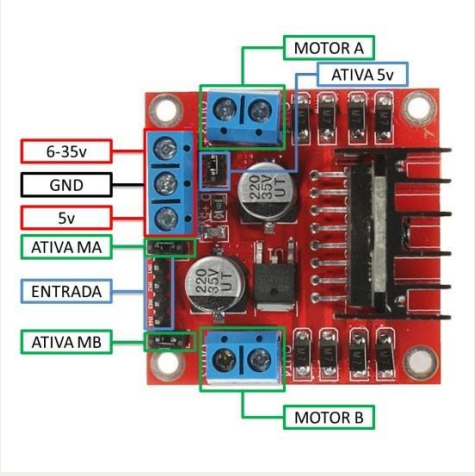


Figura 11: Circuito esquemático do funcionamento da ponte H.

Fonte: https://www.filipeflop.com/produto/driver-motor-ponte-h-l298n/

Atividade proposta

Nesta atividade, você irá realizar as conexões dos componentes que fazem seu robô interagir com o mundo com seu controlador. Fará, também, a programação da lógica que seu robô irá seguir para desviar de obstáculos.

Tente associar as funções do robô que você irá construir e programar às funções das partes do seu corpo. A qual parte do seu corpo você associaria os sensores? E os motores? E o Arduino?

3.1 Conexão física dos componentes

Para dar início à conexão dos componentes do robô, siga os passos descritos nesta seção com muita atenção e cuidado.

1 – Com a utilização de jumpers macho-fêmea, conecte os pinos do Arduino aos pinos do sensor ultrassônico, conforme a figura 12. Não é necessário utilizar as mesmas cores de jumpers mostradas no desenho, porém, é interessante utilizar, se disponíveis, o fio vermelho para a alimentação 5V e o fio preto para o GND.

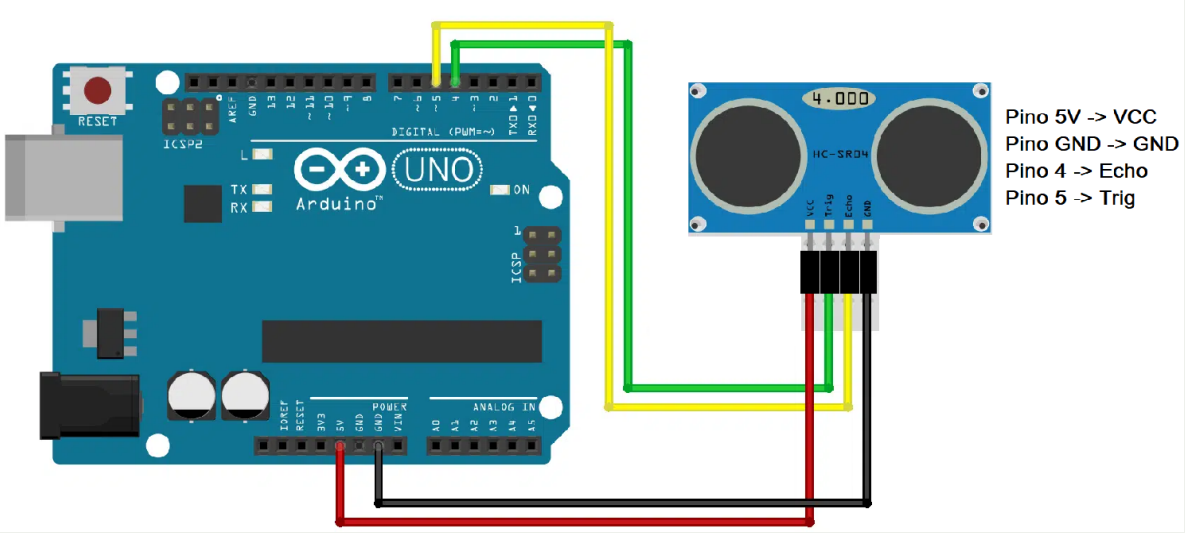


Figura 12: Conexão dos motores.
Fonte: Adaptado de http://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/

2- Utilizando jumpers, conecte os pinos do módulo Ponte H ao Arduino de acordo com a figura 13 e a tabela a seguir:

Pino no Arduino	Pino no Módulo Ponte H	Comando dos motores
Digital 6	IN1	Gira Motores da Direita para frente
Digital 7	IN2	Gira Motores da Direita para trás
Digital 8	IN3	Gira Motores da Esquerda para frente
Digital 9	IN4	Gira Motores da Esquerda para trás
Digital 10	ENA	Controla velocidade dos motores da direita
Digital 11	ENB	Controla velocidade dos motores da esquerda

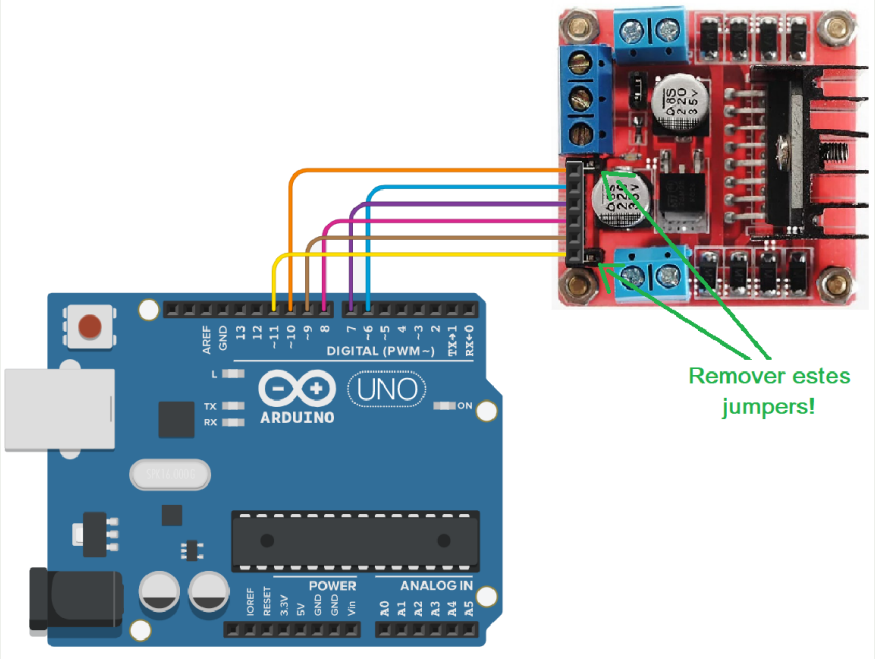


Figura 13: Conexão entre Arduino e Módulo Ponte H.
Fonte: Autoria própria

Agora que o sensor e a placa de comando dos motores já estão conectados ao controlador, é possível realizar a programação do robô, que irá determinar a maneira com que o robô realizará a sua tarefa.

3.2 Programação do Controlador Arduino – Experimento 1 – Ligar os motores

Para realizar esta atividade, você irá programar o controlador utilizando o Arduino IDE, software gratuito para programação do Arduino. A programação é transferida ao Arduino por meio da porta USB. É importante que o Arduino não esteja ligado à bateria no momento da programação.A interface do software Arduino IDE foi apresentada na Figura 6.

Inicialmente, você deverá fazer um programa mais simples, em que ligue os motores para frente o tempo todo. Para isso, você precisa:

1- Definir os pinos que serão utilizados para acionamento dos motores

```
#define IN1 6 //Pino motor A (Direita - Frente)
#define IN2 7 //Pino motor A (Direita - Trás)
```

```
#define IN3 8 //Pino motor B (Esquerda - Frente)
#define IN4 9 //Pino motor B (Esquerda - Trás)
```

2- Definir os pinos de controle de velocidade dos motores

```
#define ENA 10 //Pino velocidade motor A (Enable A)
#define ENB 11 //Pino velocidade motor B (Enable B)
```

3- Configurar a função dos pinos como SAÍDAS na função “void setup”

```
void setup() {
pinMode(IN1,OUTPUT); //Saída para motor A
pinMode(IN2,OUTPUT); //Saída para motor A
pinMode(IN3,OUTPUT); //Saída para motor B
pinMode(IN4,OUTPUT); //Saída para motor B
pinMode(ENA,OUTPUT); //Controle velocidade motor A
pinMode(ENB,OUTPUT); //Controle velocidade motor B
```

4- Ainda dentro da função setup, você deve configurar a velocidade do motor, pois isso pode ser executado apenas uma vez. Dessa maneira, a escolha da velocidade que o motor vai girar não precisa ser alterada o tempo todo. Sugere-se o valor 100 para a velocidade neste experimento, pois dessa maneira, os motores devem receber um valor de tensão de aproximadamente 3,5V, que está acima do valor mínimo (3V) e é uma velocidade baixa para começar os testes.

```
analogWrite(ENA,100); //Controle PWM do motor A (0 a 255)
analogWrite(ENB,100); //Controle PWM do motor A (0 a 255)
delay(1000); //Aguarda 1s antes de iniciar
} //fim do void setup
```

5- Após realizadas as configurações, você deve escrever o código da função “void loop”, que será executada indefinidamente, até que o Arduino seja desligado. Para que os motores sejam ligados para frente, deve-se acionar os pinos IN1 e IN3 em nível lógico ALTO e desligar os pinos IN2 e IN4, conforme a lógica a seguir:

```
void loop() //loop principal
{
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
}
```

6- Finalizada a programação, o código deve ser compilado e enviado ao Arduino. Mas antes, é importante selecionar o modelo da placa e a porta serial, no menu “Ferramentas”. Para compilar e gravar, clique no botão “Verificar” e depois “Carregar”.

Atenção: Todos os procedimentos listados neste tópico devem sem executados com a bateria desconectada do Arduino e sua alimentação será recebida pela porta USB do computador. Assim, mantenha o pino P4 do clipe de bateria desconectado do Arduino. O mesmo só deverá

ser conectado após desconectar o cabo USB, quando forem realizados testes longe do computador.

Após concluir a programação e carregar o programa no Arduino, os motores já devem ligar de maneira que as rodas se movimentem para frente. Caso as rodas estejam se movimentando em sentido diferente ou para trás, inverta a conexão no módulo Ponte H do motor que está girando no sentido contrário.

Para deixar seu código mais elegante e prático, podendo ser reutilizado mais vezes dentro do programa, você pode organizar cada parte que se repete em **uma função.** Você deve, agora, construir uma função chamada “frente”, que irá encapsular todos os comandos que você colocou na função principal no primeiro experimento. Dessa maneira, na função principal haverá apenas a chamada à função “frente. As funções devem ser construídas abaixo da função principal. O código ficará como se segue:

```
void loop() //loop principal
{
    frente();
}

void frente() //função para mover o robô para frente
{
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
} //fim da função frente
```

Após realizar essas mudanças, Compile e Carregue o programa novamente no Arduino.

Experimente, agora, definir uma função para parar o robô, enviando o nível lógico baixo para as 4 saídas digitais que comandam os motores.

```
void parar() //função para mover o robô para frente
{
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,LOW);
} //fim da função parar
```

No loop principal, faça com que o robô ande para frente por 1 segundo e pare por 1 segundo repetidamente.

```
void loop() //loop principal
{
    frente();
    delay(1000);
    parar();
    delay(1000);
}
```

3.3 Programação do Controlador Arduino – Experimento 2 - Girar

Agora que o robô já possui uma função para seguir em frente, você deve construir novas funções para fazer o robô girar para a esquerda e para a direita. Ative as saídas corretas para cada motor, de maneira que o robô gire em torno de seu eixo.

Para girar para a direita, os motores da direita devem girar para trás e os motores da esquerda devem girar para frente, ou seja, as saídas IN2 e IN3 devem ser ativadas, enquanto que as saídas IN1 e IN4 devem ser desativadas. Construa a função “direita”, completando os espaços em branco no código a seguir:

```
void direita() //função para girar o robô para direita
{
    digitalWrite(IN1,_____);
    digitalWrite(IN2,_____);
    digitalWrite(IN3,_____);
    digitalWrite(IN4,_____);
} //fim da função direita
```

Para girar para a esquerda, os motores da direita devem girar para frente e os motores da esquerda devem girar para trás, ou seja, as saídas IN1 e IN4 devem ser ativadas, enquanto que as saídas IN2 e IN3 devem ser desativadas. Construa a função “esquerda”, completando os espaços em branco no código a seguir:

```
void esquerda() //função para girar o robô para esquerda
{
    digitalWrite(IN1,_____);
    digitalWrite(IN2,_____);
    digitalWrite(IN3,_____);
    digitalWrite(IN4,_____);
} //fim da função esquerda
```

Para testar as funções, construa, na função principal, uma lógica que realize a chamada de cada uma delas. Faça com que o robô realize a seguinte sequência, com cada função sendo executada por 1 segundo: frente, parar, direita, parar, esquerda, parar. Complete os espaços em branco para chegar ao código que faz o que se pede:

```
void loop() //loop principal
{
    frente();
    delay(_____);
    ______();
    delay(1000);
    direita();
    delay(______);
    parar();
    delay(1000);
    _________();
    delay(1000);
    _______();
```

```
    delay(1000);
}
```

Agora, seu robô já está quase pronto para o desafio final – desviar de obstáculos. Ele já consegue se locomover para frente e girar, possibilitando andar para outras direções quando avistar um obstáculo. Mas ainda fala algo... ele ainda não é capaz de saber se há um obstáculo à sua frente... O que está faltando programar para que o robô possa reconhecer os obstáculos?

3.4 Programação do Controlador Arduino – Experimento Final – Desviar de obstáculos

Agora que o robô já possui as funções para se locomover em qualquer direção, é hora de programa-lo para detectar obstáculos. Essa detecção é feita por meio do sensor ultrassônico. Para utilizá-lo, você deve adicionar à IDE do Arduino a biblioteca Ultrassonic.zip (disponível em: http://downloads.arduino.cc/libraries/github.com/ErickSimoes/Ultrasonic-3.0.0.zip). Baixe o arquivo do link e inclua-o na IDE do Arduino. Para isso, abra a IDE, clique em “Sketch” > “Incluir Biblioteca” > “Adicionar Biblioteca .ZIP” e selecione o arquivo “Ultrassonic.zip”.

A conexão do sensor ultrassônico foi feita aos pinos 4 e 5 do Arduino. Por isso, é preciso definir os pinos e a instância para esse sensor no início do código, conforme as entradas conectadas fisicamente:

```
#define TRIGGER_PIN 4
#define ECHO_PIN    5
Ultrasonic ultrasonic(TRIGGER_PIN, ECHO_PIN);
```

Agora é o momento de escrever uma função que fará com que a leitura do sensor ultrassônico seja convertida em centímetros. Essa função utiliza a biblioteca instalada para emitir um sinal de ultrassom, calcular o tempo que esse sinal gasta para bater no obstáculo e voltar. Com esse tempo, a função calcula a distância do sensor ao obstáculo. Para programar essa função, utilize o código a seguir:

```
float distancia() //função que mede a distância em cm
{
      float distanciaObj;
      distanciaObj = ultrasonic.read();
      return(distanciaObj);
} //fim da função distancia
```

Neste momento chegou a hora de reformular a função *void loop* para que o robô possa seguir sempre em frente e, ao se aproximar de um obstáculo a 20 centímetros, ele

deverá parar e escolher um sentido para girar. Sugere-se que seja escolhido um sentido de giro para que o robô sempre se movimente após encontrar o obstáculo.

Uma possível solução para esse programa seria:

```
void loop()
{
  frente();
  float dist;
  dist = distancia();
    if(dist < 20)
    {
      parar();
       delay(500);
       esquerda();
       delay(400);
       parar();
       delay(500);
    }
  delay(100);
}
```

Após compilar e carregar essa solução, experimente a mudar o sentido de giro ao encontrar um obstáculo.

Desafio: elabore uma lógica em que o robô escolha aleatoriamente o sentido que ele irá girar ao encontrar um obstáculo.

Você consegue descrever outras aplicações para o robô que acabou de desenvolver?

_____

_____

_____

_____

Cursos do IFMG

Campus do IFMG	Cursos relacionados	Nível de ensino
Betim	Técnico em Automação Industrial	Ensino médio integrado
Betim	Técnico em Mecânica	Ensino médio integrado
Betim	Engenharia de Controle e Automação	Ensino Superior
Betim	Engenharia Mecânica	Ensino Superior
Congonhas	Técnico em Mecânica	Ensino médio integrado
Congonhas	Técnico em Mecânica	Ensino técnico subsequente
Conselheiro Lafaiete	Técnico em Mecânica	Ensino médio integrado
Conselheiro Lafaiete	Técnico em Mecânica	Ensino técnico subsequente
Conselheiro Lafaiete	Técnico em Eletrotécnica	Ensino médio integrado
Conselheiro Lafaiete	Técnico em Eletrotécnica	Ensino técnico subsequente
Formiga	Técnico em Eletrotécnica	Ensino médio integrado
Formiga	Engenharia Elétrica	Ensino Superior
Ibirité	Técnico em Automação Industrial	Ensino médio integrado
Ibirité	Técnico em Mecatrônica	Ensino médio integrado
Ibirité	Engenharia de Controle e Automação	Ensino Superior
Ipatinga	Técnico em Automação Industrial	Ensino médio integrado
Ipatinga	Técnico em Eletrotécnica	Ensino médio integrado
Ipatinga	Engenharia Elétrica	Ensino Superior
Itabirito	Técnico em Automação Industrial	Ensino médio integrado
Itabirito	Engenharia Elétrica	Ensino Superior
Ouro Preto	Técnico em Automação Industrial	Ensino médio integrado
Ribeirão das Neves	Técnico em Eletroeletrônica	Ensino médio integrado
Sabará	Técnico em Eletrônica	Ensino médio integrado
Sabará	Engenharia de Controle e Automação	Ensino Superior